VIDEO BASIC

20 VIDEOLEZIONI DI BASIC
PER IMPARARE COL C16

GRUPPO
EDITORIALE
JACKSON

Unità a disco
Come funziona un drive
Uso e manutenzione dei dischetti
File sequenziali e random
Istruzioni per il disco
DSAVE, DLOAD, HEADER
DIRECTORY, SCRATCH, COLLECT
Videogioco n° 14

14

C16

commodore 16 plus 4

VIDEOBASIC C 16
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore Editoriale: Roberto Pancaldi
Autore: Softidea -
Via Indipendenza 88-90 - Como
Redazione software:
Giuliano Cremonesi
Francesco Franceschini
Progetto grafico:
Studio Nuovidea - via Longhi, 16 - Milano
Impaginazione:
Moreno Confalone
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Tutti i diritti di riproduzione e pubblicazione di disegni, fotografie, testi sono riservati.
© Gruppo Editoriale Jackson 1986.
Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.p.A. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

GRUPPO EDITORIALE JACKSON
DIVISIONE GRANDI OPERE

SOMMARIO

HARDWARE **2**
Unità a disco.
Come funziona un drive.
Manutenzione dei dischetti.

IL LINGUAGGIO **10**
Il DOS. I file.
Le istruzioni per il disco,
DSAVE, DLOAD, HEADER,
DIRECTORY, SCRATCH, COLLECT

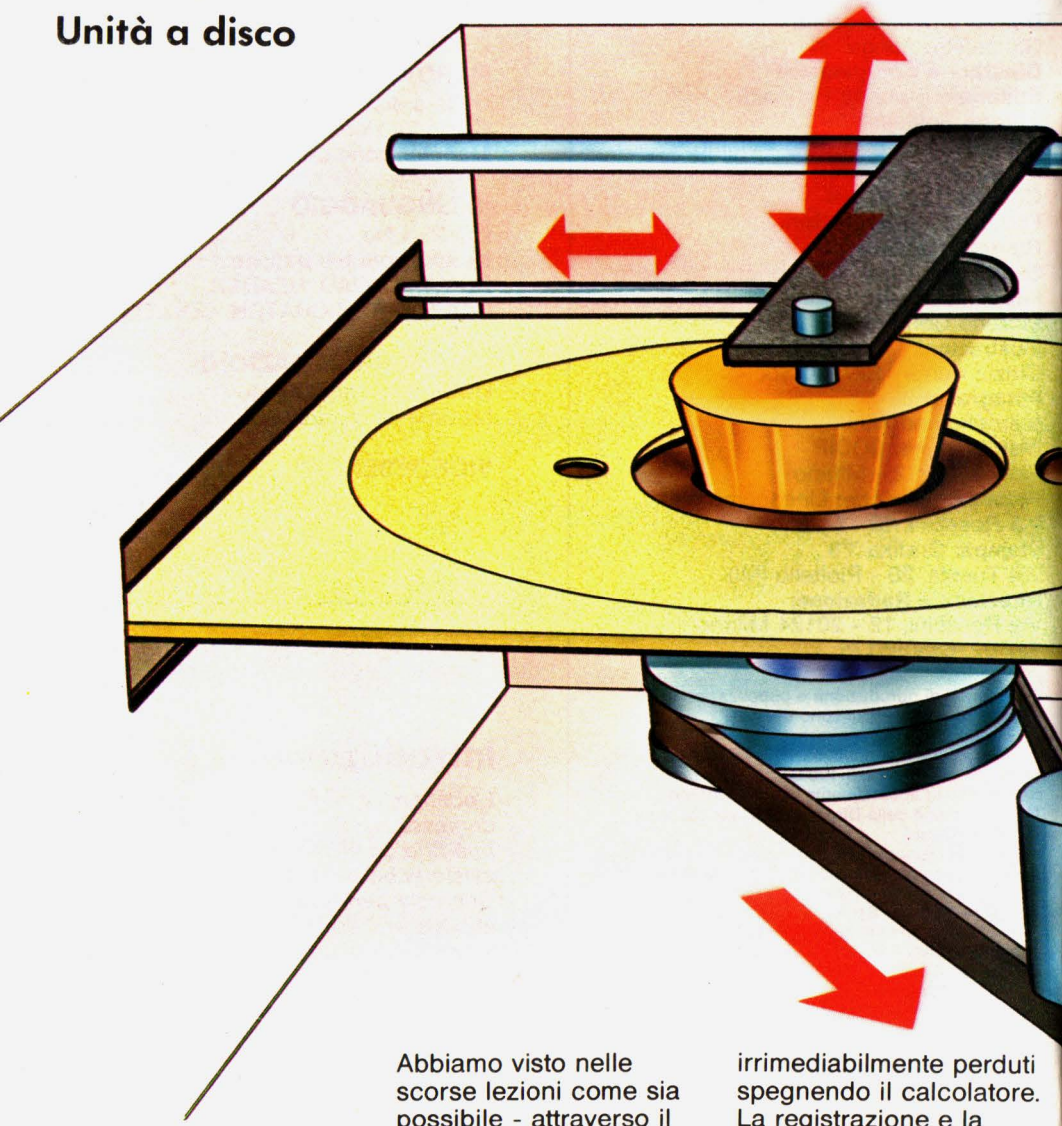
LA PROGRAMMAZIONE **20**
Usare i file sequenziali.
Movimento controllato.

VIDEOESERCIZI **32**

Introduzione

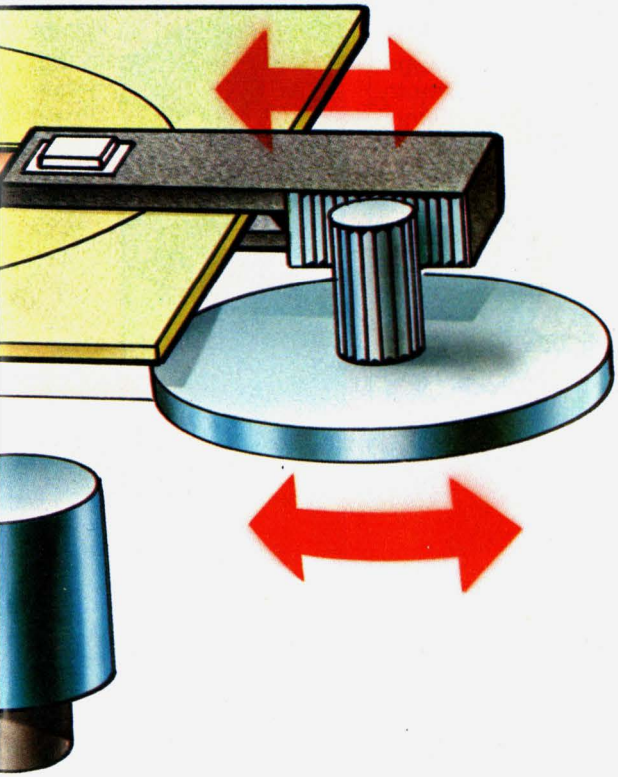
L'economico registratore è la memoria di massa per eccellenza addirittura si può dire che l'home computer non esisterebbe senza registratore.
Che noia però i lunghi e interminabili minuti per il caricamento o salvataggio di un programma, che estenuanti attese nella ricerca di un programma e che scarsa affidabilità, poi, in tutte le operazioni che vedono il registratore coinvolto.
Come sempre la tecnologia ha dato una risposta a tutti questi problemi: i floppy disk drive.
Trattare allora grosse moli di dati, in breve tempo e con una quasi totale affidabilità diventa semplice, il trucco è conoscere i comandi.

Unità a disco



Abbiamo visto nelle scorse lezioni come sia possibile - attraverso il registratore a cassette - memorizzare su nastro magnetico tutti i programmi e i dati che andrebbero altrimenti irrimediabilmente perduti spegnendo il calcolatore. La registrazione e la lettura da nastro - sicuramente avrai già avuto modo di accorgertene - è un'operazione molto

semplice, affidabile ed economica: tuttavia richiede lunghe attese per il caricamento delle informazioni e, soprattutto, necessita di un intervento manuale per avviare e spegnere i motori del registratore (il famoso "PRESS PLAY ON TAPE").



La ricerca dei programmi sul nastro è inoltre affidata a tecniche abbastanza empiriche, come il conteggio dei secondi o il rilievo del numero di giri compiuti dalla rotella di trascinamento, con il continuo rischio (soprattutto quando si è abbastanza distratti) di sovrapporre pezzi già registrati e provocare la perdita di programmi che magari avevano richiesto lunghe ore di paziente battitura.

Una delle più interessanti possibilità per l'espansione delle prestazioni offerte dal tuo C16 è allora sicuramente costituita dall'unità a dischi flessibili (floppy disk).

Lo scopo dell'unità a disco (o "disk drive"), della quale ci occuperemo nella nostra lezione, è infatti proprio quello di eliminare tutti questi "inconvenienti", permettendo una registrazione altrettanto sicura di quella offerta dal registratore a cassette, ma molto più veloce e, soprattutto, completamente automatica.

Al contrario di quando si usa il registratore, infatti,

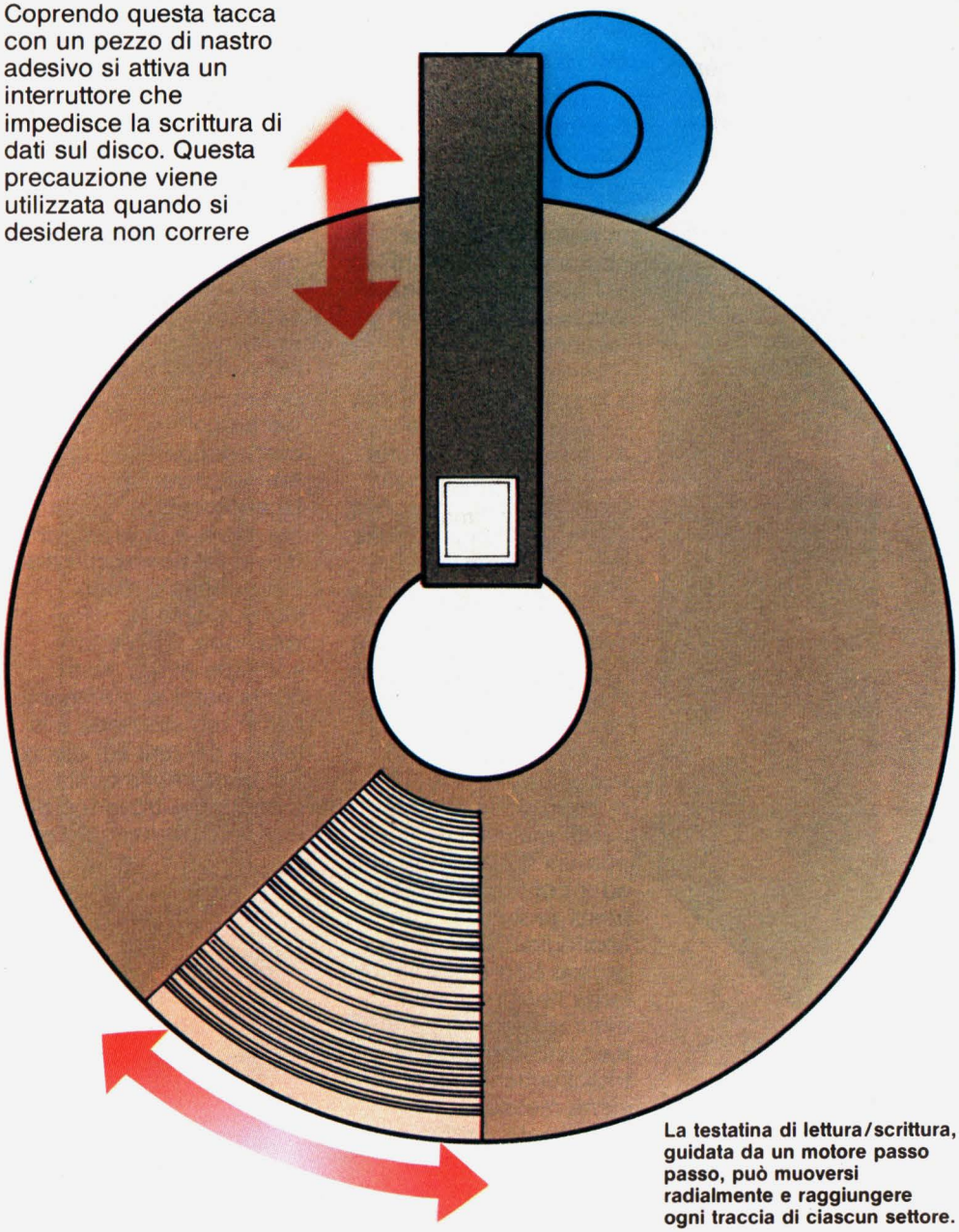
HARDWARE

non è necessario preoccuparsi in alcun modo del funzionamento del drive: la gestione delle varie operazioni - come per esempio la registrazione o il caricamento dei programmi - avviene in questo caso sotto il diretto controllo dell'unità centrale, senza più alcun intervento esterno di ricerca o di verifica.

Come funziona un drive

Il principio di funzionamento di un drive è basato su un disco magnetico, posto in rotazione da una specie di "giradischi" e sul quale una testina di lettura/scrittura può leggere o scrivere informazioni.
L'introduzione del disco all'interno dell'unità avviene attraverso una fessura, praticata nella parte anteriore del drive, immediatamente visibile date le sue dimensioni (leggermente superiori a quelle del disco: 5 pollici 1/4, circa 13 centimetri).
Il disco, chiamato anche "floppy disk" ("disco flessibile"), è costituito da un sottile foglio di materiale plastico, ricoperto con gli stessi ossidi metallici usati nei normali nastri magnetici, ed è contenuto in una busta protettiva semirigida dotata di un grosso foro e di una larga finestra. Il primo - centrale - serve per il dispositivo di trascinamento (quando viene messo in movimento il drive pone infatti in rotazione il disco; corrisponde, in un certo senso, al foro centrale dei dischi fonografici); la seconda, radiale, mette invece allo scoperto parte della superficie del disco, affinché la testina possa entrare in contatto con il supporto magnetico durante la rotazione del disco stesso.
Vicino al grande foro centrale si trova inoltre un forellino circolare attraverso il quale è possibile vedere uno o più piccoli fori corrispondenti, praticati nel disco vero e proprio. Una fotocellula, posta nel drive, è in grado di rilevare il passaggio dei fori quando il disco viene messo in rotazione. Questi fori vengono allora usati come punti di riferimento per la registrazione e la lettura dei dati sul disco.
Sul bordo della busta protettiva si trova inoltre una tacca rettangolare.

Coprendo questa tacca con un pezzo di nastro adesivo si attiva un interruttore che impedisce la scrittura di dati sul disco. Questa precauzione viene utilizzata quando si desidera non correre



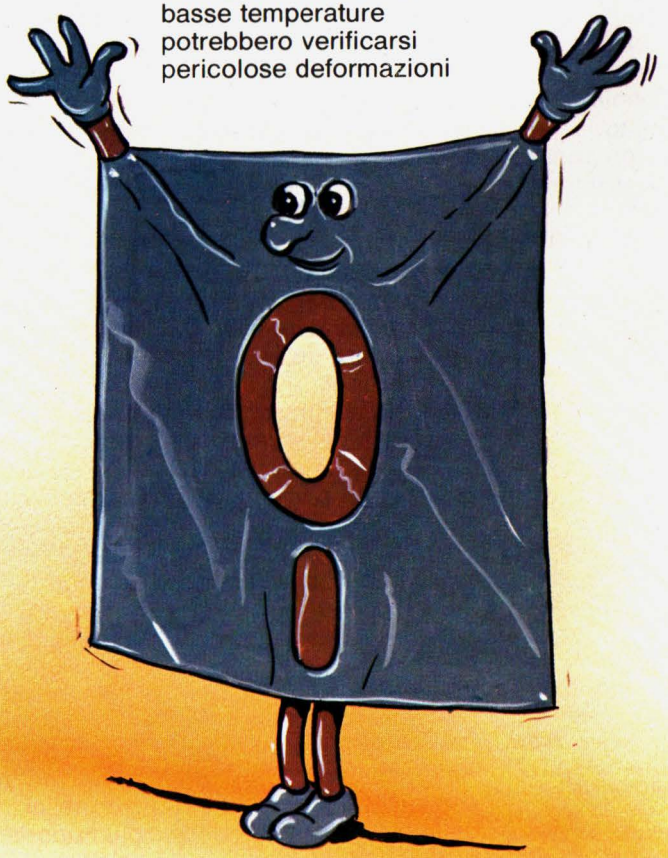
La testatina di lettura/scrittura, guidata da un motore passo passo, può muoversi radialmente e raggiungere ogni traccia di ciascun settore.

rischi di cancellazioni o alterazioni accidentali dei dati sul disco.
All'interno del drive sono presenti i motori per la rotazione del disco e per lo spostamento della testina, oltre naturalmente all'insieme dei circuiti necessari per pilotarli ed all'interfaccia per comunicare - attraverso il cavetto di collegamento - con l'unità centrale.
Il sistema di controllo dei dischi è inoltre costituito in modo da definire su ciascun dischetto vergine (cioè non ancora registrato) una struttura complessa, ma esattamente definita. Il floppy disk, infatti, quando è nuovo non è immediatamente utilizzabile: su di esso devono essere preventivamente registrate alcune informazioni indispensabili per il suo successivo funzionamento.
Affinché possa leggere un disco il drive richiede quindi che lo stesso abbia subito la cosiddetta "formattazione", cioè un'operazione che registri sul disco stesso una sorta di "mappa geografica" che la testina di lettura/scrittura utilizzerà come sistema di riferimento per orientarsi nella registrazione o nella lettura delle informazioni.
Senza addentrarci troppo in inutili dettagli tecnici, è sufficiente sapere che durante la formattazione il disco viene suddiviso in una serie di invisibili tracce concentriche, le quali sono a loro volta ulteriormente scomposte nei cosiddetti settori, che prendono questo nome proprio perché sono rappresentabili esattamente come settori delle circonferenze corrispondenti alle tracce.
Complessivamente, alla fine della formattazione, il disco risulta costituito da 683 settori ciascuno dei quali in grado di contenere 256 byte. Di questi 683 settori, solo 664 sono tuttavia effettivamente utilizzabili (gli altri 19 servono infatti al drive per memorizzarvi alcune informazioni a suo esclusivo uso e consumo): la reale capacità di memorizzazione di un dischetto è quindi pari a circa 170000 byte (664 * 256).

Manutenzione dei dischetti

Abbiamo visto che, nonostante le ridotte dimensioni, un floppy disk è in grado di registrare con precisione una grandissima quantità di informazioni; per questa ragione è quindi di importanza capitale osservare scrupolosamente alcune semplici, ma indispensabili precauzioni, analoghe a quelle già viste a proposito dei nastri magnetici, ma ancora più importanti vista la relativa delicatezza dei

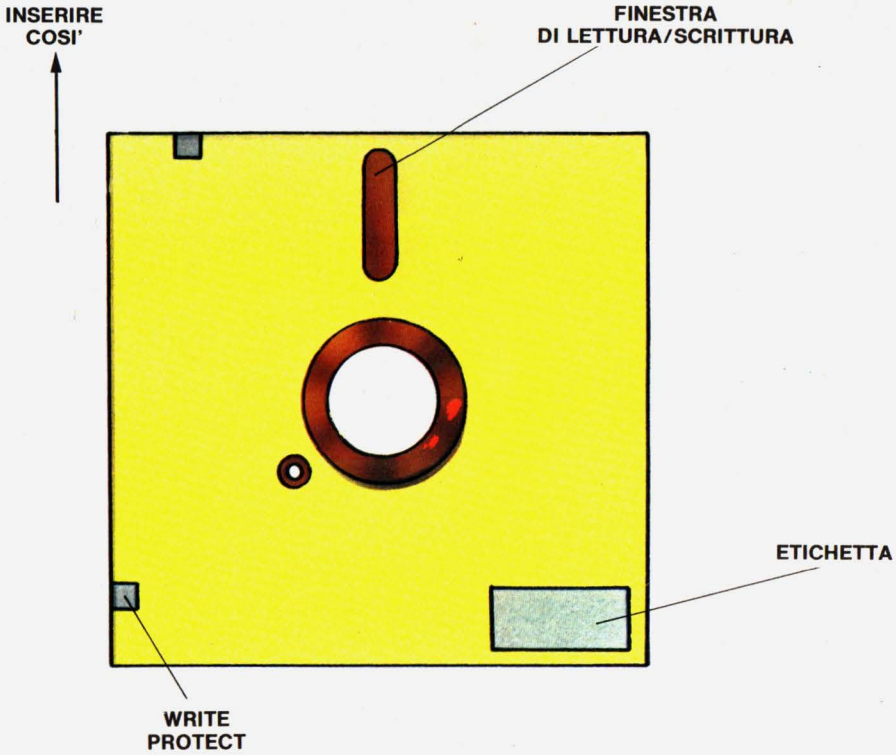
dischetti:
- non toccare mai la superficie magnetica dei dischetti (attraverso il foro ovale). Sulle dita è sempre presente un sottile strato di grasso, che può impedire il corretto contatto tra testina e dischetto;
- evitare che la polvere si depositi sui dischetti: anche un microscopico granello di polvere o di cenere di sigaretta può infatti rovinare irrimediabilmente lo strato magnetico. La soluzione migliore è quella di riporre il dischetto nella propria busta di protezione non appena terminato di usarlo;
- non esporre i dischetti al sole, al caldo o alle basse temperature potrebbero verificarsi pericolose deformazioni del supporto plastico;
- per quanto il termine "disco flessibile" possa aprire la strada alle più disparate interpretazioni, evitare nel modo più assoluto di sollecitare meccanicamente i dischi, curando di non maneggiarli con eccessiva sicurezza, di



non sovrapporli e di non schiacciarli con pesi di alcun genere.

- allontanare il più possibile i dischi da calamite, televisori, altoparlanti e in generale da qualunque apparecchiatura elettrica;
- tenere una copia di riserva di tutti i dischetti contenenti informazioni importanti, conservandola in un luogo diverso da quello in cui si trovano gli originali;
- quando il computer è spento non lasciare mai il dischetto nel drive: estrarlo sempre prima di spegnere la macchina e riporlo immediatamente.

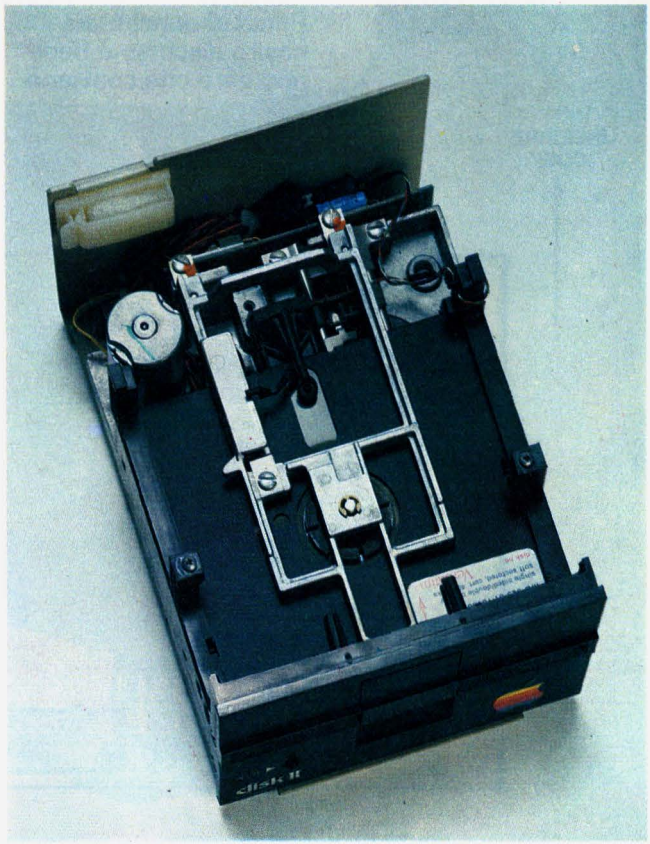
Prima di concludere il nostro discorso è bene ricordare che conviene sempre acquistare dischetti di buona marca a doppia densità di registrazione (doppia densità significa che la qualità del supporto magnetico è in grado di rispondere nel migliore dei modi alle più piccole sollecitazioni impartite dalla testina del drive, con la quasi assoluta sicurezza e garanzia di non perdere -a causa di eventuali difetti dovuti alla grossezza della grana- alcun dato o informazione).



Il DOS

Adesso che ti sei fatto un'idea del funzionamento dell'unità a dischi, è necessario fermarci un momento per parlare di come possa avvenire lo scambio delle informazioni tra l'elaboratore e il disk-drive.
La gestione dell'unità a dischi è affidata a un particolare programma, chiamato DOS (abbreviazione di Disk Operating System, sistema operativo del disco), che si occupa di coordinare le molteplici attività ed operazioni eseguibili dal drive.
La funzione del DOS è sotto certi aspetti molto simile a quella dell'interprete BASIC, dato che permette di impartire e di controllare

Interno di un'unità a disco.



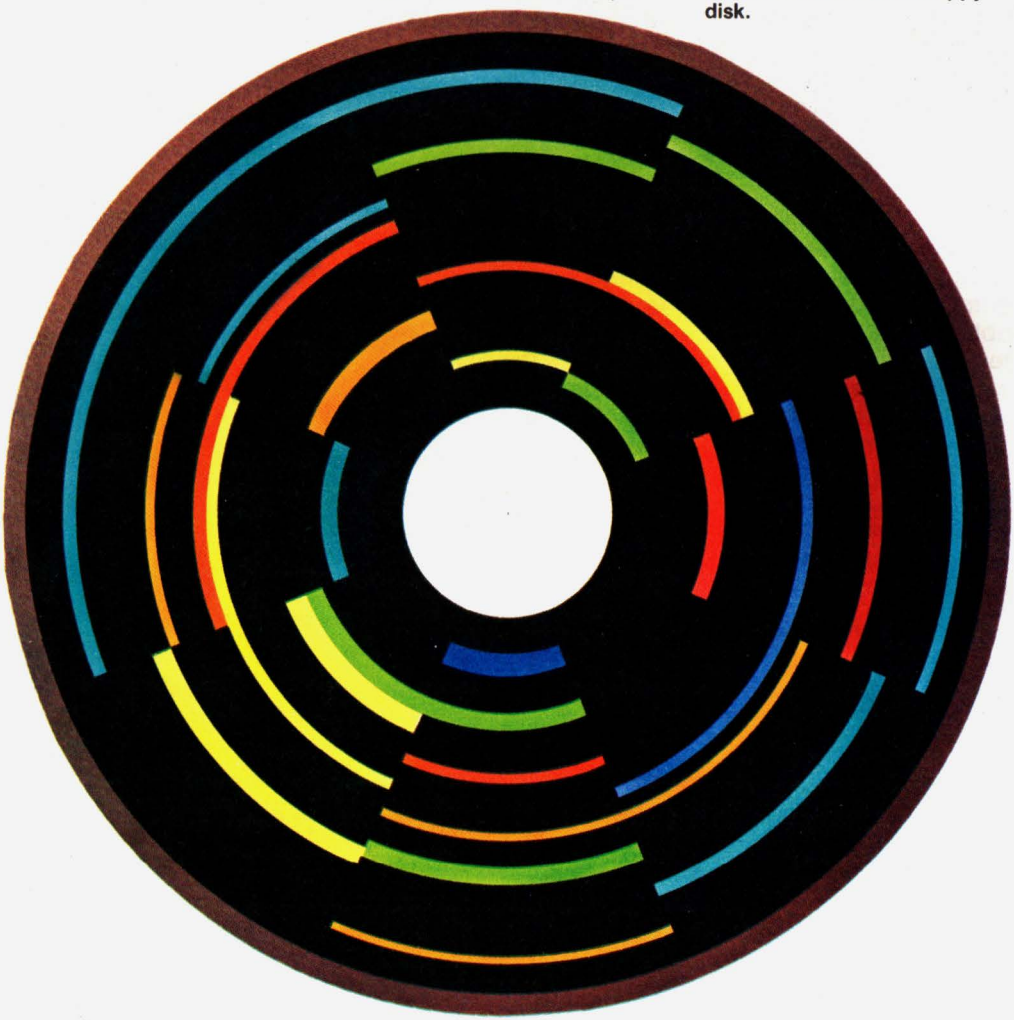
- attraverso pochi e brevi comandi di facile comprensione - operazioni tutt'altro che semplici, facendosi quindi carico delle parti più noiose e ripetitive. In effetti, l'intero funzionamento di un elaboratore è sempre completamente gestito da un sistema operativo, che controlla e coordina in continuazione (senza che l'utilizzatore abbia la minima possibilità di accorgersene) il funzionamento di tutto il sistema.
Nel caso specifico il DOS è la parte di sistema operativo che si occupa dell'unità a dischi.
Per quanto riguarda il C16 il DOS risiede in una zona di memoria ROM posta all'interno del disk-drive; la parte di sistema operativo, residente nella memoria centrale, che deve gestire i dischi è quindi ridotta alla sola interfaccia, di tipo seriale, con l'unità a dischi: tutto quanto serve alla effettiva gestione del drive è eseguito all'interno dell'unità stessa.
All'unità centrale resta solo il compito di comunicare i comandi all'unità periferica e leggere le risposte.
Quando si dà un comando per il drive viene trasmesso sul cavo di collegamento seriale; il drive lo riconosce ed inizia ad eseguirlo. Nel caso in cui il comando non implichi ulteriori scambi tra unità centrale e periferica l'unità centrale, una volta accertata la ricezione del comando, continua l'esecuzione del programma in corso; in caso contrario attende le informazioni in risposta dal drive. Se si tenta di inviare un altro comando al disco, questo viene eseguito solo al termine del comando precedente, poiché la ricezione stessa del comando risulta bloccata fintanto che l'unità periferica è impegnata nell'esecuzione di un altro comando.
Questo schema di procedimento, che permette di migliorare le prestazioni di tutto l'insieme, è completamente controllato dal DOS. Più avanti impareremo i comandi riconosciuti dal DOS.

I file

I programmi sono abbastanza utili, ma, diciamo la verità, quello che realmente si vuole dal calcolatore è trattare informazioni, dati, numeri, nomi, indirizzi, quantità.
Qualunque istruzione che possa essere scritta può anche essere inserita nella memoria del computer e, di conseguenza, memorizzata sul drive.
Un file è un insieme di informazioni che, per un motivo qualunque, si desidera raggruppare.
Esempi di file possono essere: i nomi degli alunni di una classe, i numeri estratti durante una partita di tombola, i nomi e gli indirizzi dei nati in un certo giorno, un programma. Il DOS mette a disposizione principalmente due tipi di file:
— file sequenziali: sono file in cui le informazioni vengono registrate una di seguito all'altra, in modo sequenziale. Sono i più semplici da utilizzare, ma non permettono una grande flessibilità di uso;
— file random (ad accesso diretto): in questo caso le informazioni sono scritte (o lette) in un punto qualsiasi del file.
Rispetto al file sequenziale il file random è più pratico ed efficiente, ma ha una struttura leggermente più rigida e quindi più complicata da gestire.
Esamineremo quindi come dovremo fare per memorizzare su disco - raccogliendoli tra loro - tutti i dati e le informazioni che desideriamo non siano irrimediabilmente perduti spegnendo il calcolatore.
Per fare questo è comunque inizialmente necessario introdurre due nuovi termini, usatissimi quando si parla di file: Record e Campi.
Facciamolo (per essere più chiari possibile) attraverso un esempio.
L'elenco telefonico della tua città, preso nel suo insieme, è un file: esso contiene infatti informazioni ordinate e classificate secondo un ordine ben preciso.
Questo file è formato da migliaia di record, ciascuno dei quali costituito da un cognome, un nome, un indirizzo e un numero di telefono. Un record può essere quindi definito come un gruppo di dati correlati fra loro; le singole registrazioni che si susseguono nel

record si chiamano invece campi.
Nel nostro caso esistono 4 campi per ciascun record: un campo Cognome, un campo Nome, un campo Indirizzo e un campo Numero di telefono.
La scelta del formato del record (cioè del numero di campi) è naturalmente arbitraria: avremmo potuto considerare - oltre ai campi appena visti - anche un campo Numero di prefisso, indicante appunto il prefisso telefonico.
Abbiamo detto che nei file sequenziali (come dice il nome stesso) le informazioni sono

Settori e tracce su un floppy disk.



registrate una di seguito all'altra. Ciò significa che per leggere (o scrivere) l'ultima informazione del file occorre aver prima letto (o scritto) tutte le informazioni precedenti. Questa limitazione può sembrare un notevole svantaggio (sarebbe alquanto scomodo se anche noi per cercare "Rossi Mario" nell'elenco telefonico dovessimo cominciare dalla prima pagina), tuttavia - grazie alla notevole velocità di lettura dei dati sul drive - entro file di dimensioni abbastanza contenute il tempo di ricerca di un singolo record è abbastanza ragionevole. Verificheremo e approfondiremo comunque il nostro discorso nella parte della lezione dedicata alla programmazione.

Le istruzioni per il Disk Drive

Il C16 ed il PLUS 4 dispongono di un gruppo completo di istruzioni BASIC capaci di gestire l'unità a dischi in modo molto efficiente. Tra queste potrai notarne alcune del tutto simili alle corrispondenti dedicate al registratore, altre invece del tutto nuove, in grado di soddisfare le più ampie possibilità offerte da questa periferica.
In realtà il C16 nasce orientato all'uso dei dischi. Se premi infatti i tasti funzione F2, F3, e F5, attiverai dei comandi per il disk drive.

DSAVE

Lo puoi ottenere anche premendo F5; serve a salvare sul disco posto all'interno del drive il programma in memoria in quel momento.
È indispensabile assegnare al programma un nome (max 16 caratteri) da scrivere di seguito al comando tra virgolette.

```
DSAVE "TOMBOLA 1"
```

salva su disco il programma in memoria col nome TOMBOLA 1.

Sintassi dell'istruzione

DSAVE "NOME PROGRAMMA"

DLOAD

Puoi evitarne la scrittura premendo F2; carica da disco nella memoria RAM del tuo computer il programma il cui nome deve essere specificato dopo il comando, racchiuso tra virgolette.

DLOAD "TOMBOLA 1"

carica da disco il programma di nome TOMBOLA 1.

Sintassi dell'istruzione

DLOAD "NOME PROGRAMMA"

HEADER

Ogni dischetto nuovo, per poter essere utilizzato deve essere sottoposto ad una procedura di preparazione chiamata formattazione.
HEADER è il comando in grado di svolgere questo compito.

HEADER "DISCOGIORGIO", D0,IAA

procede alla formattazione di un dischetto il cui nome sarà DISCOGIORGIO sul drive numero 0 (si possono infatti collegare più drive contemporaneamente) con Identificazione AA.
HEADER, usato con un disco contenente dati e programmi, ne provoca la cancellazione totale, esattamente come se si trattasse di un disco nuovo. Occore perciò prestare la massima attenzione.

Sintassi del comando

HEADER "NOMEDISCO",DN,IID

DIRECTORY

Questo comando, attivato anche tramite tasto funzione F3, visualizza su video l'elenco dei programmi contenuti su un disco senza rovinare l'eventuale programma in memoria in quel momento.
Data la grande capacità di un dischetto, l'elenco può superare il numero di righe dello schermo. In questo caso lo scrolling può essere fermato con i tasti Ctrl e S. Il tasto [←] serve invece a rallentare la stampa delle informazioni.

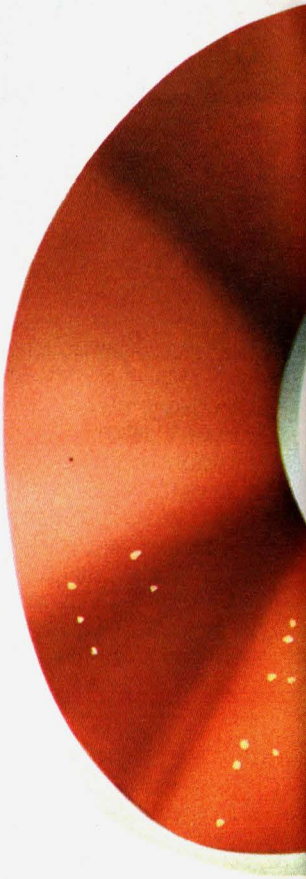
LINGUAGGIO

Sintassi del comando

DIRECTORY

SCRATCH

Può accadere che un programma o dei dati presenti su un disco siano più interessanti e per questo, per far posto a cose più importanti, sia conveniente cancellarli. SCRATCH è il comando che permette di eliminare il file specificato e di recuperare lo spazio da questo occupato.



SCRATCH "TOMBOLA 1"

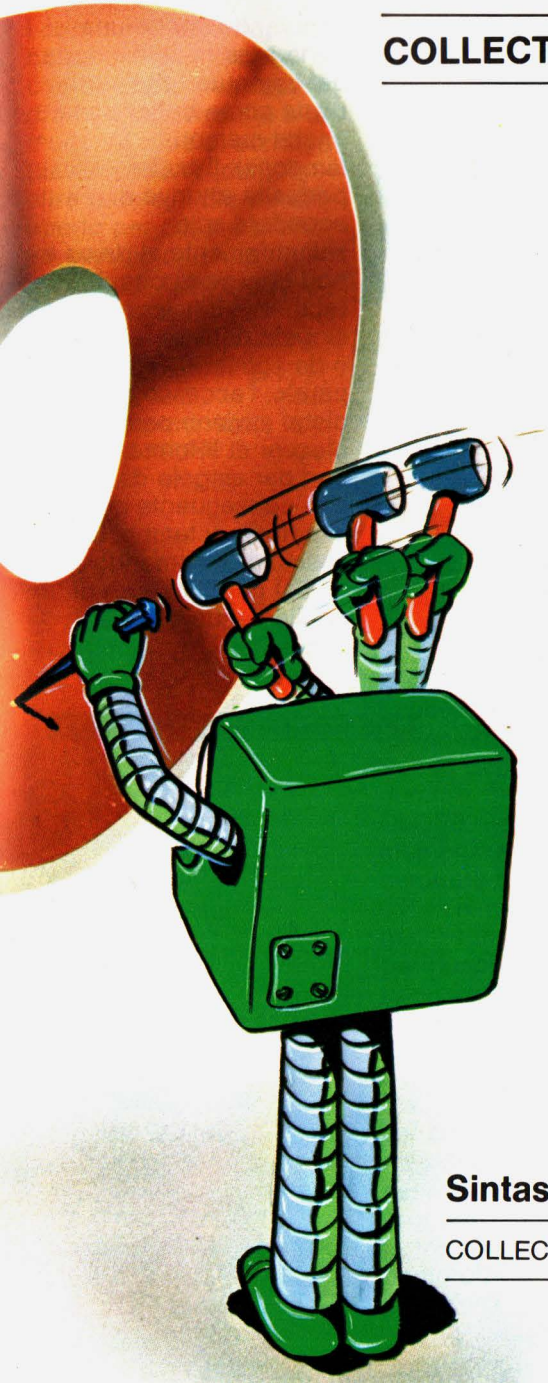
Cancella il programma TOMBOLA 1 dal disco posto nel drive. Poiché una volta eseguito il comando non sarà più possibile il recupero dei dati cancellati, il sistema chiede una ulteriore conferma prima di procedere alla cancellazione con il messaggio "ARE YOU SURE?" (SEI PROPRIO SICURO?); rispondendo Y il file sarà calcellato, altrimenti l'operazione sarà annullata.

Sintassi del comando

SCRATCH "NOMEPROGRAMMA"

COLLECT

Dopo aver effettuato delle cancellazioni su un disco e ogni qualvolta serva <ripulirlo> è buona norma utilizzare questo comando. COLLECT, infatti, è in grado di sistemare un disco dove sono presenti file non chiusi correttamente e di aggiornare la DIRECTORY e la RAM eseguendo infine una operazione di convalida.



Sintassi del comando

COLLECT

Usare i file sequenziali

In questa parte della lezione approfondiremo il discorso che poco fa avevamo cominciato a proposito dei file.
Innanzitutto, prima di poter accedere a un file bisogna aprirlo.
Aprire un file significa comandare al DOS di ricercare delle informazioni relative a questo file: se esso è già sul disco e, se lo è dove si trova esattamente.
Il comando OPEN fa eseguire al drive tutte queste operazioni, riservando anche una certa area nella memoria del computer che verrà usata come buffer (detta anche memoria tampone). Il buffer permette al drive di accedere solo saltuariamente al disco e non ogni volta che il programma ha un singolo dato da leggere o da scrivere. In altre parole, l'accesso al disco avviene per blocchi di informazioni e non per singolo dato, con conseguente grande risparmio di tempo. Un esempio di comando OPEN per l'apertura di un file sequenziale è il seguente:

```
OPEN 2, 8, 2, "PROVA, S, W"
```

I caratteri S e W dopo il nome del file indicano che questo dovrà essere di tipo S(equenziale) e di modo W, cioè di scrittura (dall'inglese W (rite)). In altre parole, il file viene aperto per essere scritto se avessimo voluto aprirlo per leggerne il contenuto avremmo dovuto mettere, anziché W, una R (R(eading) in inglese).
A questo punto tutte le istruzioni PRINT # che si riferiranno al canale appena aperto provocheranno la memorizzazione sul disco. Questo breve programma illustra il funzionamento di quanto abbiamo appena detto:

```
10 OPEN 2, 8, 2, "PROVA,
   S, W"
20 FOR I = 1 TO 200
30 LET A = RND (0)
40 PRINT A
50 PRINT # 2, A
60 NEXT I
70 CLOSE 2
```

Dopo l'apertura del file verranno inviati 200 numeri casuali rispettivamente al video e al drive; alla fine del ciclo il file sarà chiuso e i numeri saranno stati memorizzati sul disco. Subito dopo l'ultimo numero, in seguito alla CLOSE, verrà inoltre memorizzato un carattere di EOF (End OF File, cioè fine del file) necessario al computer per riuscire a gestire in futuro quel file, quando vorrai recuperare ciò che vi era scritto.
L'importanza di CLOSE, oltre che per la scrittura del carattere EOF, è rilevabile anche osservando questo programma al lavoro. Potrai infatti renderti conto che la spia rossa del drive non rimarrà accesa in continuazione: le informazioni verranno infatti registrate solo quando il buffer sarà stato man mano riempito nel corso dell'esecuzione. Se non avessimo inviato la CLOSE, i dati rimasti, contenuti nel buffer al termine del ciclo, non avrebbero mai raggiunto

il dischetto, visto che il computer si sarebbe aspettato nuovi "arrivi" ed il file sarebbe risultato incompleto. Quindi occorre sempre ricordarsi di chiudere i file.
Vediamo adesso come fare per recuperare i numeri che abbiamo appena memorizzato sul disco. Anziché inviare delle PRINT, ora dovremo richiedere i dati con delle INPUT:

```
10 OPEN 2, 8, 2, "PROVA, S, R" : REM FILE DI
   LETTURA
20 FOR I = 1 TO 200
30 INPUT # 2, A
40 PRINT A
50 NEXT I
60 CLOSE 2
```

Il funzionamento del programma sarà identico a quello visto prima, tranne che il computer - anziché scrivere - leggerà dal disco.
Un'importanza particolare rivestono, al momento della scrittura dei file, i caratteri separatori tra i vari campi. Per poter leggere i dati con l'istruzione INPUT è infatti necessario che i campi siano registrati, facendo comparire dopo ciascuno di essi un carattere separatore valido, cioè riconosciuto come tale dal disco. I caratteri separatori riconosciuti sono il RETURN (CHR$ (13)) e la virgola. Il primo può essere inserito con la funzione CHR$ (13), oppure facendo terminare la istruzione PRINT senza punteggiatura (come abbiamo fatto noi nell'esempio appena visto).
Il secondo provoca invece delle limitazioni, che causano talvolta perdita di dati o insorgere di errori: poiché richiede una certa esperienza per essere utilizzato, ti conviene non usarlo mai. La cosa migliore, senza entrare troppo in inutili

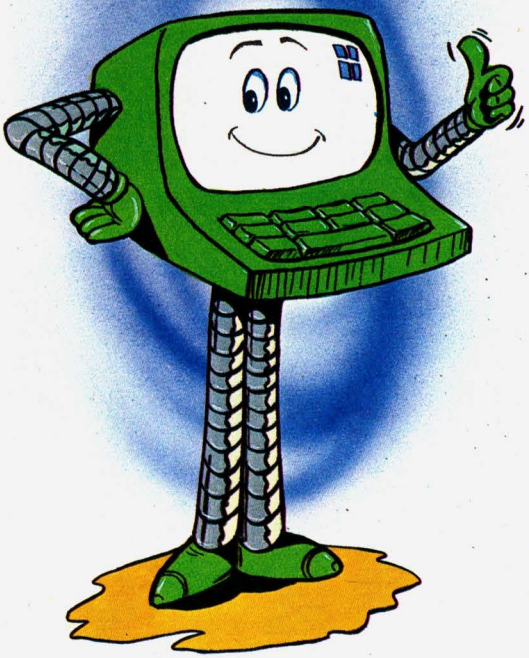
dettagli, è quella di non eseguire mai PRINT od INPUT multiple (cioè del tipo PRINT # 4, A, B, C o INPUT # 4, A, B, C), ma ricorrere sempre a più istruzioni semplici. Per quanto i programmi diventino leggermente più lunghi, la facilità di uso di una simile soluzione permette di restare sempre padroni della situazione, almeno sino a quando non si è acquistata sufficiente familiarità con i segreti dei file.

Un'altra cosa da segnalare è che l'aggiornamento di un file sequenziale può essere ottenuto soltanto riscrivendo completamente il file. Infatti la presenza del carattere EOF, inserito subito dopo l'ultimo record, impedisce l'inserimento di nuovi elementi in fondo alla lista.
Le uniche cose da fare sono allora:

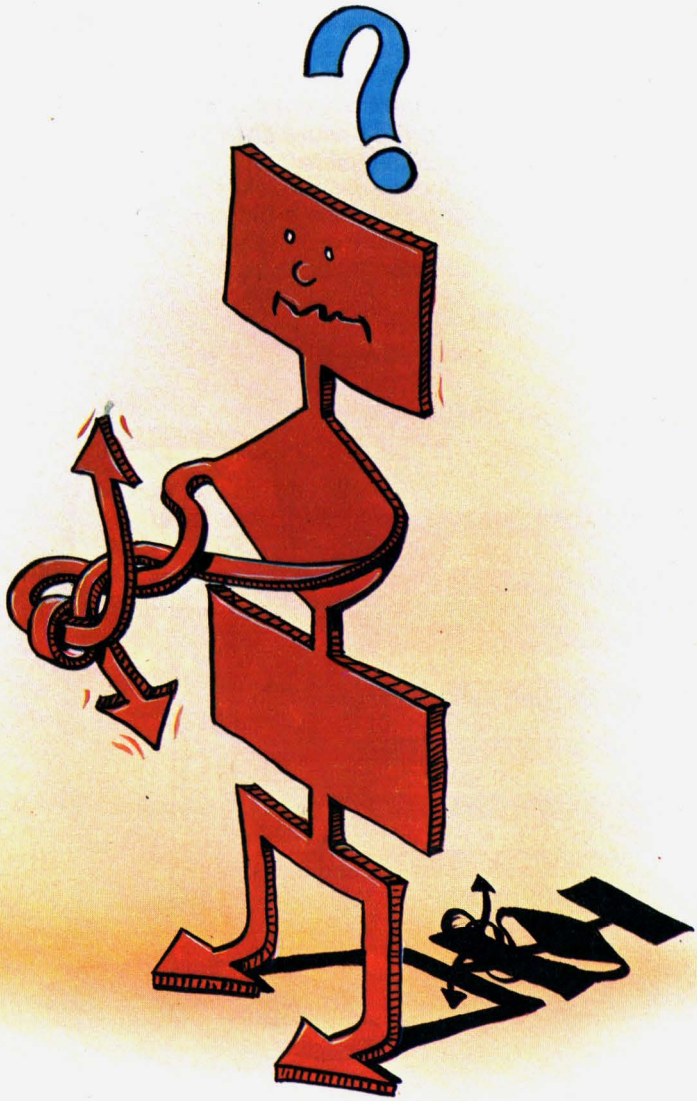
— aprire in lettura il file da aggiornare;
— aprire in scrittura un nuovo file;
— copiare ordinatamente i record dal vecchio file nel nuovo apportando via via le modifiche, le cancellazioni o le aggiunte;
— chiudere alla fine i due file, cancellando il vecchio e cambiando il nome al nuovo file, attribuendogli il nome vecchio.
Vediamo quindi come avremmo dovuto fare per aggiungere altri 100 numeri casuali al file che avevamo creato prima:

```
 10 OPEN 2, 8, 2, "PROVA, S, R"
 20 OPEN 10, 8, 10, "PROVA2, S, W"
 30 FOR I = 1 TO 200
 40 INPUT # 2, A
 50 PRINT # 10, A
 60 NEXT I
 70 FOR I = 1 TO 100
 80 PRINT # 10, RND (0)
 90 NEXT I
100 CLOSE 10 : CLOSE 2
110 SCRATCH "PROVA"
120 RENAME "PROVA 2" TO "PROVA"
130 END
```

L'esempio che abbiamo appena fatto era abbastanza semplice, visto che sapevamo perfettamente quanti numeri dovevamo trasferire dal vecchio al nuovo file prima di passare alla estrazione di nuovi numeri casuali. Il più delle volte il numero di letture che possono essere effettuate prima di arrivare all'EOF è invece sconosciuto: tieni inoltre presente che cercare di leggere oltre l'ultimo record provoca un messaggio di errore. Non è quindi possibile leggere i record "a casaccio".
Un metodo quasi infallibile è allora il seguente. Bisogna

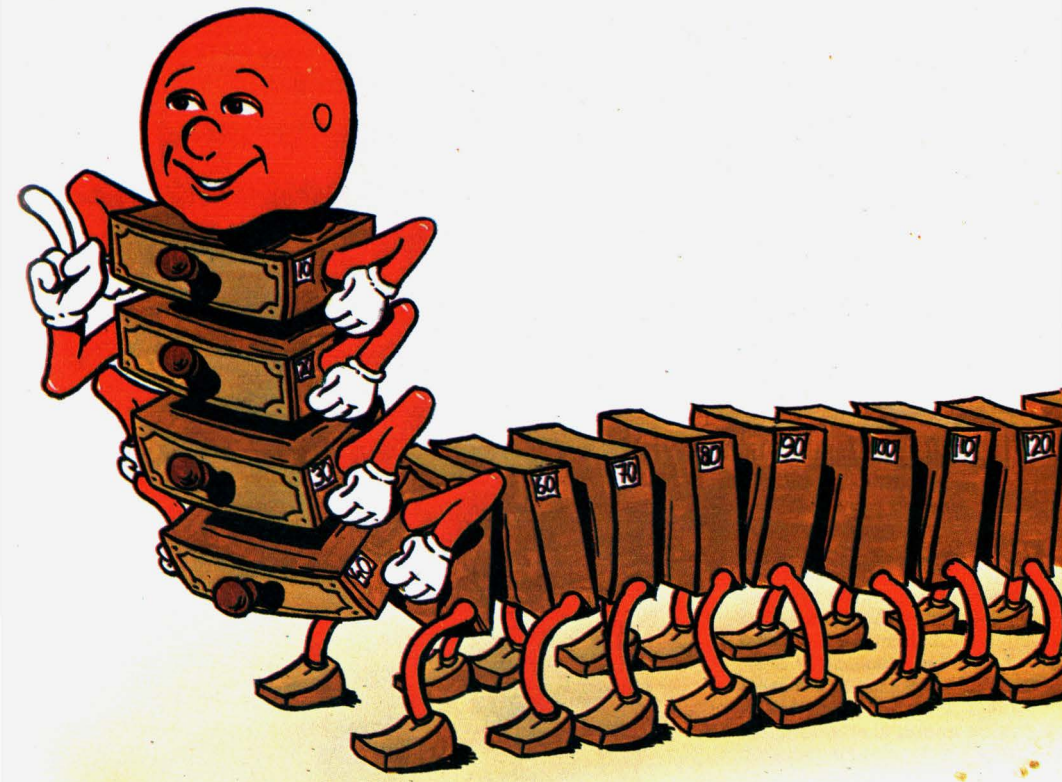


modificare il programma, in modo da scrivere nel file, come ultimo record, uno o più caratteri speciali, o facilmente riconoscibili, che tu stabilirai essere il segno di fine-file (per esempio con il nostro file di numeri avremmo potuto



scrivere un numero negativo; con un file di stringhe potremmo invece scrivere una stringa di segnalazione con un significato inequivocabile, come "Z * Z * Z").

In questo modo in fase di lettura basterà inserire una istruzione di controllo di fine file, che risulterà verificata al momento in cui il carattere speciale verrà incontrato dal programma.



Ecco un esempio in proposito. Il programma che segue scriverà sul disco un file, composto da tante stringhe quante ne vorrai introdurre dalla tastiera. Quando deciderai di terminare, basterà che tu batta solo RETURN.
Prima della chiusura del file il computer scriverà automaticamente come ultimo record il nostro carattere speciale "Z * Z * Z".

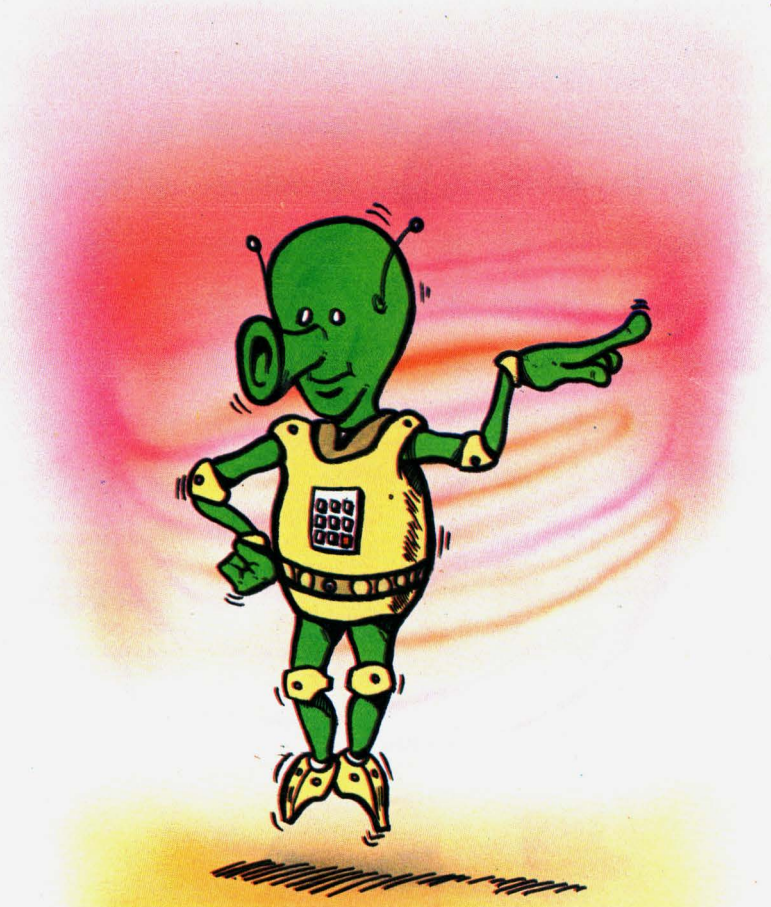
```
10 OPEN 2, 8, 2,"PROVA3, S, W"
20 INPUT "SCRIVI LA STRINGA (SOLO RETURN PER FINIRE)"; A$
30 IF A$ = "*" THEN GOTO 60
40 PRINT # 2, A$
50 GOTO 20
60 PRINT # 2,"Z * Z * Z"
70 CLOSE 2
```

L'errore è sempre in attesa: malignamente aspetta una tua distrazione, anche minima. Non dargli mai la possibilità di sogghignare.
Impara la sintassi delle istruzioni.
Realizza programmi in forma modulare, facilmente leggibili, e molto documentati.
Nella digitazione, poi metti la massima attenzione: anche una virgola al posto sbagliato, compromette la buona riuscita del programma.

Potrai adesso recuperare e rileggere le stringhe che hai appena battuto, con questo secondo programma:

```
10 OPEN 2, 8, 2, "PROVA3, S, R"
20 INPUT # 2, A$
30 IF A$ = "Z * Z * Z" THEN CLOSE 2 : END
40 PRINT A$
50 GOTO 20
```

Adesso la conoscenza del numero di elementi appartenenti al file non è più necessaria: il nostro "Z * Z * Z" avverte infatti l'unità centrale di non tentare di leggere oltre, perché non esistono altri record. In questo modo ci siamo messi in salvo da un'eventuale possibilità di errore.



Movimento controllato

Con questo programma introduciamo una delle tecniche per controllare un carattere o una animazione per mezzo della tastiera (tasti cursore).
Dalla riga 35 alla riga 60 si effettuano le operazioni di verifica del tasto premuto e si determinano gli spostamenti relativi.

```
 10 SCNCLR:POKE56,56:POKE55,0:CLR
 20 COLOR 0,1:COLOR 4,1:COLOR 1,1
 30 POKE 65299,(PEEK(65299)AND3)+56:POKE 65298,PEEK(65298)AND 251
 40 FOR I = 0 TO 7
 50 READ A:POKE 14336+I,A:POKE 14592+I,0:NEXT I
 60 LET X=0:LET Y=0:PRINT " BE @"
 70 GET KEY T$
 80 LET T=ASC(T$):LET C=32=GOSUB 150
 90 IF T=17 AND Y<22 THEN LET Y=Y+1
100 IF T=29 AND X <39 THEN LET X=X+1
110 IF T=145 AND Y>0 THEN LET Y=Y—1
120 IF T = 157 AND X>0 THEN LET X=X—1
130 IF T =83 THEN POKE 65299,(PEEK 65299)AND 3)+208:POKE65298,
    PEEK(65298)ORG:END
140 LET C= 0:GOSUB 150 : GOTO 70
150 LET A = X + Y * 40 : POKE 3072 + A, C
160 POKE 2048 + A, 7 * 16 + 1 : RETURN
170 DATA 24, 24, 60, 126, 189, 60, 102, 195
```

VIDEOESERCIZI

Dopo aver esaminato attentamente il listato rispondi alle domande verificando le tue risposte personalmente alla tastiera.

```
10 GOSUB 50
20 GOSUB 10
30 RETURN
40 END
50 S=32400:S=S+1
60 PRINT "QUANTE VARIABILI..."
70 GOTO 30
80 IF S<56 THEN RETURN
90 RETURN
```

Quale errore causa questo programma?

La "IF" in linea 80 potrà mai essere vera?

Il programma si ferma con la "END" ... o no?

```
100 K=K+1:GOSUB 100:REM PER CONTARE
    QUANTI GOSUB OCCORRONO PER RIEMPIRE LO STACK
```

Quanti GOSUB può sopportare il "GOSUB STACK"?

GRUPPO
EDITORIALE
JACKSON